

# Transferring Learning To Rank Models for Web Search

Craig Macdonald<sup>1</sup>, B. Taner Dincer<sup>2</sup>, Iadh Ounis<sup>1</sup>

<sup>1</sup> University of Glasgow, Glasgow G12 8QQ, UK

<sup>2</sup> Dept of Statistics & Computer Engineering, Mugla University, Mugla, Turkey

{craig.macdonald, iadh.ounis}@glasgow.ac.uk<sup>1</sup>, dtaner@mu.edu.tr<sup>2</sup>

## ABSTRACT

Learning to rank techniques provide mechanisms for combining document feature values into learned models that produce effective rankings. However, issues concerning the transferability of learned models between different corpora or subsets of the same corpus are not yet well understood. For instance, is the importance of different feature sets consistent between subsets of a corpus, or whether a learned model obtained on a small subset of the corpus effectively transfer to the larger corpus? By formulating our experiments around two null hypotheses, in this work, we apply a full-factorial experiment design to empirically investigate these questions using the ClueWeb09 and ClueWeb12 corpora, combined with queries from the TREC Web track. Among other observations, our experiments reveal that ClueWeb09 remains an effective choice of training corpus for learning effective models for ClueWeb12, and also that the importance of query independent features varies among the ClueWeb09 and ClueWeb12 corpora. In doing so, this work contributes an important study into the transferability of learning to rank models, as well as empirically-derived best practices for effective retrieval on the ClueWeb12 corpus.

**Categories and Subject Descriptors:** H.3.3 [Information Storage & Retrieval]: Information Search & Retrieval

**Keywords:** Learning-to-rank, Web Search

## 1. INTRODUCTION

Learning to rank [18], a means of utilising machine learning techniques in information retrieval (IR), provides a way to combine various features - such as query dependent weighting models and query independent document features - into effective *learned models*, based on learning from existing queries and associated relevance judgments. For the purposes of effective retrieval, a learned model can then be used to effectively (re)rank the documents retrieved for any given new query, by measuring the same features on the given query and the retrieved documents [22].

A learning to rank technique is typically trained on the same corpus for which the learned model is to be applied.

However, there are situations when there may be a need to *transfer* the learned model to another (e.g. newer) corpus, or if the learned model obtained from a *subset* of the target corpus can be applied on the target corpus. Both situations arise within participations to the TREC Web track [5]: for example, do learned models obtained when training upon the category B subsets generalise to the larger category A sets of the ClueWeb09 or ClueWeb12 corpora; orthogonally, do learned models obtained using ClueWeb09 transfer to the newer ClueWeb12? Next, the features that can be employed by a learning to rank technique can vary greatly in kind, as well as in number (e.g. 44 query dependent and independent features are used by the LETOR datasets [19], while the MSLR datasets released by Microsoft<sup>1</sup> have 136 features per document). Hence, it is important to understand the role of the types of features present in the learned model, and how these learned models encapsulating different features transfer between corpora.

In this study, we consider a state-of-the-art learning to rank technique, namely LambdaMART [3], in comparison with another technique based on linear learning for experimental purposes [24], and examine two null hypotheses of interest through an empirical study conducted upon the ClueWeb09 and ClueWeb12 document corpora, as used by the TREC Web tracks 2009-2012 and 2013-2014.

Indeed, this paper contributes the first large-scale study in transfer learning for Web search, by means of a full-factorial experiment design, to provide empirically supported observations concerning three aspects: within-corpus training, cross-corpus training, and choice of feature set. Among many results, we show the importance of query independent quality features for the ClueWeb09 corpus - in contrast to ClueWeb12 - which is due to the high prevalence of spam documents within the older ClueWeb09. Hence, our experiments and analysis provide new insights for researchers and practitioners into the transfer of learning to rank models between corpora and how the corpus used to train/test can impact upon the choice of effective features (c.f. field-based query dependent weighting model features), while also providing empirically derived best practices for effective retrieval upon the TREC ClueWeb12 corpus.

The remainder of this paper is structured as follows: In Section 2, we introduce the necessary background concerning learning to rank; Section 3 defines our experimental setup and evaluation methodology; Our experimental results are described in Section 4; We provide concluding remarks in Section 5.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

ICTIR'15, September 27-30, 2015, Northampton, MA, USA.

© 2015 ACM. ISBN 978-1-4503-3794-6/15/10 ...\$15.00.

DOI: <http://dx.doi.org/10.1145/2808194.2809463>.

<sup>1</sup><http://research.microsoft.com/en-us/projects/mslr/>

## 2. BACKGROUND

Learning to rank techniques have become an oft-deployed technology that can combine many feature values into an effective ranking of documents. The output of the (offline) learning phase is a *learned model*, which defines how feature values should be combined to make the final predicted relevance scores for each ranked document. Learned models may take the form of the weights for a linear combination of feature values (e.g. AFS [24]) or of regression trees defined on feature values (e.g. GBRT [33] or LambdaMART [3]). When ranking documents for an unseen query, the learned model is applied on the computed feature values to produce the final ranking of documents for the query.

The *candidate document set*<sup>2</sup> is an important aspect during learning to rank: typically, a single standard weighting model selects a number of top-ranked documents for each query to later re-rank. These documents then have feature values computed. Liu [18] suggests that a single standard weighting model such as BM25 is sufficient, but not the most effective, while Macdonald et al. [22] showed that the size of the candidate set should approach 5000 documents for ranking Web documents, and for mixed types of information needs it should not consider anchor text. Dang et al. [10] showed that applying proximity to create the candidate set could improve effectiveness. Finally, we note that the candidate set should be created using an identical ranking approach during both offline learning and online ranking - this prevents selection bias, which can occur if relevant documents are artificially inserted into the candidate set used for learning [22, 26].

Transfer learning – the act of transferring knowledge from a *source domain* to a *target domain* – has been investigated within machine learning (e.g. [28] for an overview). Different scenarios exist [29]: *feature transfer* refers to the scenario when there is not full overlap between the features in the source and target domains; on the other hand, when feature sets do overlap, *instance-based transfer* allows training data from the source and target domains to train a learner for the target domain. Techniques (e.g. [9, 15]) in the latter scenario focus on the appropriate selection and combining of training instances from source and target domains to create appropriate models. Instance-based transfer techniques are not naturally applicable to learning-to-rank scenarios. Indeed, the effective learning-to-rank techniques classically encompass a listwise component, and hence selecting at the level of instances (documents) within the candidate sets for a query may hinder accurate calculation of listwise loss functions. For this reason, our study addresses the direct transfer of learning-to-rank models. Later, in Section 4, we describe and use a transfer learning technique that is directly applicable to learning-to-rank. We therefore leave the adaptation of instance-based transfer techniques for future work.

Relatedly, to encourage research within transfer learning for learning to rank, the Yahoo! 2011 Learning to Rank Challenge included a transfer learning task. This tasked participants to examine how different learning to rank techniques could be trained given a large amount of training data on one corpus, and a lesser amount on a different *target* corpus. In particular, the challenge provided two datasets, one from a US sample of the web, and one from an Asian sample with less training data [4]. In contrast, in this work, we examine the extent to which a learned model can be effec-

tively transferred between different samples of the English web, in particular, between smaller and larger corpus subsets, as well as between older and newer Web corpora, when both are represented using identical feature sets. Additionally, the impact on the importance of different types of feature between the corpora are examined. In the next section, we pose two null hypotheses that we address through later experimentation using the ClueWeb09 and ClueWeb12 corpora and 250 queries from the TREC 2009-2013 Web track campaigns.

## 3. RESEARCH METHODOLOGY & EXPERIMENTAL SETUP

Our experiments are structured around the the general problem of transferability within learning to rank, namely how learned models transfer within different subsets of a corpus, and across corpora. We address these research investigations through experiments with systematic variations of dependent variables, as detailed below. For inferential purposes, the research investigations are formulated into two null hypotheses that follow common practices within the learning to rank community. By showing that these two hypotheses – defined below – can be rejected, we contribute new knowledge and understanding of the deployment of learning to rank.

Firstly, we investigate how learned models can be transferred between training and test corpora, as follows:

**NULL HYPOTHESIS 1.** *The retrieval effectiveness of a learning to rank technique that is trained on the target corpus will be higher than the retrieval effectiveness of the same technique when it is trained on a corpus other than the target corpus.*

This null hypothesis will allow to determine whether different forms of “other” corpora can provide more effective learned models than can be obtained from the target corpus itself. In particular, we investigate two types of “other” corpus: (i) a subset of the target Web corpus – which we call *within-corpus* training, and (ii) an older corpora also sampled from the larger Web, which we call *cross-corpus* training.

Moreover, we address the importance of feature sets for an effective learned model, and how this relates to the training and test corpus within a transfer learning scenario, formulated as follows:

**NULL HYPOTHESIS 2.** *The contribution of different feature sets to the retrieval effectiveness of a learning to rank technique are uniform across corpora.*

In posing this null hypothesis, we argue that the default scenario for a practitioner faced with a new corpus (without training data) would be to consider that the importance of different types of features within a learned model for Web search would be consistent across corpora, and hence they would train only on the old corpus. By empirically investigating this null hypothesis, we can ascertain whether the importance of feature sets change between corpora, and how this affects the effectiveness of the resulting learned models.

In the context of the above null hypotheses, we consider the systematic variation of four factors: (1) training corpus, (2) test (target) corpus, (3) learning to rank technique, and (4) feature set. We apply a *full-factorial* design – i.e. testing all combinations of factors – in order to collect empirical

<sup>2</sup>Also known as the sample in [18, 22].

**Table 1: Factors and corresponding levels considered in our experiment.**

Factors	Levels	Code
Training	ClueWeb09 Cat A	<b>cw09a</b>
	ClueWeb09 Cat B	<b>cw09b</b>
	ClueWeb12 Cat A	<b>cw12a</b>
	ClueWeb12 Cat B	<b>cw12b</b>
Test	ClueWeb09 Cat A	<b>CW09A</b>
	ClueWeb09 Cat B	<b>CW09B</b>
	ClueWeb12 Cat A	<b>CW12A</b>
	ClueWeb12 Cat B	<b>CW12B</b>
LTR Method	LambdaMART	<b>LM</b>
	AFS (linear)	<b>LIN</b>
Feature Sets	Weighting Models	<b>WM</b>
	Field Models	<b>FM</b>
	QI Features	<b>QI</b>
	Proximity	<b>PX</b>

data as parsimoniously as possible while providing sufficient information to make dependable estimates of the effects of the four factors on retrieval effectiveness. To systematically vary the factors, we assign each factor a discrete set of levels as given in Table 1. The instantiations of these factors are explained in the remainder of this section.

Full factorial designs measure response variables (in our case ERR@20, the official measure for recent TREC Web track campaigns, as per [5]) using every treatment (combination of the factor levels). A full factorial design for  $n$  factors with  $N_1, \dots, N_n$  levels requires  $N_1 \times \dots \times N_n$  experimental runs - one for each treatment. Note that we follow an ablation approach when factoring out the levels of feature sets, in that groups of features are *removed* from a total of 64 commonly applied query dependent and query independent features. For example, NoQI denotes All features minus the query independent features. Therefore, there are 160 combinations of the factor levels in total.

### 3.1 Corpus and Topics

Our experiments are conducted using open test collections created within the context of the Text REtrieval Conference (TREC). In particular, while various learning to rank datasets exist (such as LETOR, MSLR & Yahoo! Learning to Rank Challenge), we build our own learning-to-rank datasets based upon the TREC test collections for several reasons: the chosen TREC test collections represent identical retrieval tasks upon two different large English Web corpora (and subsets thereof); the features are identically formulated between the different corpora; and finally the features can be explained within the paper<sup>3</sup>.

Specifically, we use the ClueWeb09 and ClueWeb12 corpora of Web documents, used by the TREC Web tracks between 2009-2012 [5] and 2013 [6], respectively. In particular, we use two subsets of each corpus: the full category A, which contains all documents identified as being written in English - this is about 500M for ClueWeb09 and 730M for ClueWeb12; and category B, which contains a smaller number of English documents. In particular, the category B subset of ClueWeb09 represents a *controlled* subset of the category A corpus - similar to the first tier index of a com-

mercial Web search engine [32] - in that the 50M selected documents exhibit higher crawl priority. Indeed documents identified earlier within Web crawls are more likely to be relevant because of their higher quality [27]. On the other hand, the category B of ClueWeb12 is built to be a *random* sample, with no higher likelihood of including relevant documents. It is built by selecting every 17th document from the category A corpus, and is  $\sim 7\%$  of the size of category A. As will be seen in Section 4, these contrasting methodologies for the category B subsets impacts upon the results obtained for within-corpus transfer.

For ClueWeb09, there are 200 TREC topics from the 2009-2012 Web tracks with corresponding relevance assessments, graded with labels 1 - 4. Similarly, there are 50 topics from the 2013 Web track with relevance assessments within ClueWeb12. For each topic, relevant documents can be from both the category A or category B subset of each corpus. Moreover, for each corpus, we split the available topics to obtain a fair 5-fold cross validation, with each fold containing 3 parts training, 1 part validation (used by the learning to rank techniques to set parameters such as number of iterations) and 1 part testing topics. Hence, we can conduct 5-fold cross validation separately on each corpus.

Next, we note that this experimental setting allows fair cross-comparisons of the transfer of learned models between older and newer corpora, and between subset of corpora, both within cross-validation setting. For instance, the effectiveness of learned models obtained from the older ClueWeb09 on the newer ClueWeb12 target corpus can be compared with those obtained directly on ClueWeb12 training data. Similarly, the effectiveness of learned models from category B subsets of ClueWeb09 and ClueWeb12 can be contrasted with those obtained directly on the target category A sets.

To explain how these comparisons are achieved and following Table 1, we firstly denote our nomenclature: cw09a and cw09b (cw12a and cw12b) denote learned models trained using the category A and category B subsets of ClueWeb09 (ClueWeb12); in contrast, we denote the corpus used to evaluate/test a learned model in capitals, e.g. CW09A and CW09B. Note that because of a cross-fold validation setting, the testing of a learned model upon the same corpus as it is trained on is clearly possible while maintaining separation of training and testing topics: CW09A(cw09a). Moreover, the converse - CW09B(cw09a) - is also valid, where a learned model is trained on category A, but tested using the feature values and relevance assessments for category B - we call this within-corpus training. In doing so, we can compare the effectiveness of a learned model trained on the same corpus or on a different subset of the corpus. Finally, cross-corpus evaluation can be conducted, e.g. CW12A(cw09a). For such a setting, each test fold, say fold 1, would be evaluated on topics from the test part on the target corpus (CW12A), but using a model learned upon topics from the training and validation parts of the fold 1 of the training corpus, cw09a.

Finally, following past TREC Web tracks [5], we use ERR@20 to measure effectiveness.

### 3.2 Retrieval System

We conduct experiments using the Terrier IR platform [20], making use of the “fat” framework [23] to efficiently generate document rankings with multiple query dependent features suitable for applying learning to rank. Hence, using Ter-

<sup>3</sup>In contrast to the Yahoo! Challenge dataset, where the definitions of the features have not been released.



**Table 2: Feature sets applied for both category A and category B.**

Code	Features	Total
(Candidate Set)	BM25	1
WM	Weighting models on the whole document [23] (DFree, DPH [1], PL2 [1], BM25 [31], LM, MQT [22], LGD, DFIC [11, 17], DFIZ [11, 17])	8
FM	Weighting models as above on each field, namely: title, URL, body and anchor text; + PL2F [21]	37
PX	Term-dependence proximity models (MRF [25], pBiL [30])	2
QI	URL (e.g. length) link (e.g. inlink counts, PageRank) & content quality (e.g., fraction of stopwords, table text [2], spam classification [7]) features	16
TOTAL		64

rier v4.0, we index the documents of the CW09A, CW09B, CW12A and CW12B corpora, while considering the body, the title, the URL, and the anchor text from the incoming hyperlinks to be separate fields of each document.

At retrieval time, for each query, we use a light version of Porter’s English stemmer, and – following [8] – rank 5000 documents to form the *candidate document set* for each query, which will later be re-ranked by the learned models. We use BM25 [31] to generate the candidate set, as Liu [18] reports this is sufficient for effective retrieval, without considering anchor text, which results in the candidate sets with the highest recall of relevant documents. Indeed, a candidate set size of 5000 documents and use of a representation without anchor text follows from the recommendations in [22].

Upon the candidate document set generated by BM25, we add a further 63 features, which represent common query dependent and query independent features implemented by previous studies [22] and datasets such as LETOR [19] and MSLR. We categorise the features as per Table 1: various effective weighting models (WM) computed on the whole document [23]; field models computed on each field individually (FM) [19, 23]; proximity features (PX), which score highly documents where the query terms occur closely together; and query independent (QI) features to identify high quality documents, based on quality, URL and link evidence. All 64 features are summarised in Table 2.

Finally, we deploy two learning to rank techniques, which differ in the form of their learned model. The first is Automatic Feature Selection (AFS) [24], which greedily selects the next feature that most improves effectiveness upon the training set, and adds it to the currently selected features, with a feature weight selected using simulated annealing [16]. We denote this learning to rank technique as LIN in Table 1, as its learned model takes the form of a linear combination of feature values. For our second technique, we use the state-of-the-art LambdaMART technique (denoted LM in Table 1), which forms a learned model of gradient boosted regression trees [3]. Within such a learned model, the feature values for each document define a path through a decision tree, which produces the outcome value. Many such trees form the final learned model. A LambdaMART implementation won the 2011 Yahoo! Learning to Rank challenge [4]. We use the Jforests implementation<sup>4</sup> of LambdaMART.

<sup>4</sup><https://code.google.com/p/jforests/>

**Table 4: Average ERR@20 scores, factored out w.r.t. training corpus (rows) and test corpus (columns), over 2 learning to rank techniques and 4 feature sets. Values on the diagonal represent Null Hypothesis 1, while † denotes a significant difference from the diagonal result (paired t-test,  $p \leq 0.025$ ).**

Training Corpus	Test Corpus				Grand Average
	CW09B	CW12B	CW09A	CW12A	
cw09b	<b>0.1740</b>	0.1097	0.1288†	0.1481	0.1401
cw12b	0.1287†	0.1140	0.0987†	<b>0.1611</b>	0.1256
cw09a	0.1633†	0.1055	<b>0.1527</b>	0.1503	<b>0.1429</b>
cw12a	0.1341†	<b>0.1241</b> †	0.1108†	0.1595	0.1321
Grand Average	0.1500	0.1113	0.1227	0.1547	0.1352

## 4. RESULTS

Table 3 reports the ERR@20 effectiveness of the main experiments performed in this paper, including grand averages across test corpora, feature sets, and learning to rank techniques. For example, from the CW09B column (1st column) of Table 3 we observe that training on the same corpus (cw09b) is most effective for all possible combinations of feature sets considered.

Moreover, to aid in the analysis of the various factors in Table 3, Figure 1 summarises the effectiveness as a factor interaction plot. In particular, in the figure, each factor (training corpus, test corpus, learning to rank technique and ablated feature set) varies for each row and column of the plot, with legends on the diagonal. The legends apply to all graphs in the same row. For instance, from the graph in the second column of the first row, we can similarly observe that the effectiveness of learned models tested on CW09A are highest when they are trained on the same corpus (cw09a), over all learning to rank techniques and feature sets. In the remainder of this section, we make use of Table 3 and Figure 1, as well as appropriate summary tables and figures depicting multiple comparison tests, to firstly address Null Hypothesis 1 for within-corpus transfer (Section 4.1) and cross-corpus transfer (Section 4.2), respectively, before addressing Null Hypothesis 2 in Section 4.3.

### 4.1 Within-Corpus Training

To facilitate the addressing of Null Hypothesis 1, concerning the choice of training corpus, Table 4 shows the *average* ERR@20 scores obtained by factoring out w.r.t. training corpus and test corpus over all learning to rank techniques and feature sets. We firstly note that if Null Hypothesis 1 holds true for within-corpus transfers of learned models, we would expect the highest effectiveness for each test corpus to occur when the corresponding training corpus is used, (e.g. CW09A(cw09A)), which occur on the first diagonal of Table 4. To test whether Null Hypothesis 1 is true, we can perform significance tests comparing the diagonal effectiveness scores with the corresponding scores in the same column listed for the other training corpora.

On analysis of the table, we observe that the highest effectiveness scores (emphasised) occur on the first diagonal for only two test corpora. In particular, for CW09B, training on cw09b provides a significantly more effective learned model than that obtained from cw09a (according to a two-tailed paired *t*-test with a *p*-value less than 0.025). Similarly, for CW09A, training on cw09a is significantly more effective

**Table 3: Average ERR@20 scores factored out w.r.t. feature set and training corpus (rows) vs. test corpus and learning to rank techniques (columns). Recall that: cw09a, cw09b, etc. denote training corpora; CW09A, CW09B, etc. denote test corpora; ALL, NoWM, etc. are ablated feature sets; and LIN and LM denote the AFS and LambdaMART learning to rank techniques, respectively.**

		CW09B			CW12B			CW09A			CW12A			Grand
		LIN	LM	Average	LIN	LM	Average	LIN	LM	Average	LIN	LM	Average	
ALL	(Average)	0.1438	0.1609	0.1523	0.1204	0.1144	0.1174	0.1263	0.1270	0.1267	0.1688	0.1539	0.1614	0.1395
	cw09b	0.1806	0.1800	0.1803	0.1135	0.1023	0.1079	0.1296	0.1375	0.1335	0.1582	0.1405	0.1494	0.1428
	cw12b	0.1198	0.1391	0.1294	0.1158	0.1162	0.1160	0.0912	0.0967	0.0940	0.1719	0.1615	0.1667	0.1265
	cw09a	0.1696	0.1670	0.1683	0.1216	0.1078	0.1147	0.1706	0.1576	0.1641	0.1726	0.1455	0.1591	0.1516
	cw12a	0.1052	0.1575	0.1313	0.1308	0.1314	0.1311	0.1140	0.1163	0.1151	0.1724	0.1680	0.1702	0.1369
NoWM	(Average)	0.1607	0.1556	0.1581	0.1143	0.1175	0.1159	0.1286	0.1234	0.1260	0.1620	0.1475	0.1547	0.1387
	cw09b	0.1884	0.1709	0.1796	0.1130	0.1026	0.1078	0.1525	0.1292	0.1409	0.1655	0.1304	0.1480	0.1441
	cw12b	0.1356	0.1307	0.1331	0.1056	0.1229	0.1142	0.0953	0.1050	0.1001	0.1658	0.1558	0.1608	0.1271
	cw09a	0.1794	0.1761	0.1777	0.1089	0.1118	0.1104	0.1575	0.1656	0.1615	0.1564	0.1415	0.1489	0.1496
	cw12a	0.1394	0.1446	0.1420	0.1296	0.1329	0.1313	0.1093	0.0939	0.1016	0.1603	0.1623	0.1613	0.1340
NoFM	(Average)	0.1465	0.1603	0.1534	0.1128	0.1003	0.1065	0.1397	0.1318	0.1357	0.1630	0.1465	0.1547	0.1376
	cw09b	0.1811	0.1764	0.1787	0.1093	0.1082	0.1087	0.1319	0.1165	0.1242	0.1629	0.1411	0.1520	0.1409
	cw12b	0.1131	0.1511	0.1321	0.1184	0.1109	0.1146	0.1158	0.1280	0.1219	0.1588	0.1594	0.1591	0.1319
	cw09a	0.1636	0.1626	0.1631	0.1039	0.0813	0.0926	0.1690	0.1563	0.1626	0.1707	0.1375	0.1541	0.1431
	cw12a	0.1281	0.1512	0.1397	0.1198	0.1006	0.1102	0.1422	0.1263	0.1342	0.1595	0.1480	0.1537	0.1345
NoQI	(Average)	0.1311	0.1395	0.1353	0.1128	0.1033	0.1081	0.1029	0.1015	0.1022	0.1526	0.1408	0.1467	0.1231
	cw09b	0.1437	0.1562	0.1500	0.1164	0.0897	0.1030	0.1041	0.1097	0.1069	0.1391	0.1280	0.1336	0.1234
	cw12b	0.1094	0.1301	0.1197	0.1044	0.1118	0.1081	0.0826	0.1017	0.0922	0.1598	0.1548	0.1573	0.1193
	cw09a	0.1415	0.1417	0.1416	0.1065	0.0929	0.0997	0.1221	0.1139	0.1180	0.1534	0.1176	0.1355	0.1237
	cw12a	0.1299	0.1300	0.1300	0.1239	0.1189	0.1214	0.1028	0.0806	0.0917	0.1580	0.1628	0.1604	0.1259
NoPX	(Average)	0.1463	0.1554	0.1508	0.1213	0.1161	0.1187	0.1229	0.1232	0.1231	0.1652	0.1472	0.1562	0.1372
	cw09b	0.1803	0.1824	0.1813	0.1175	0.1245	0.1210	0.1280	0.1489	0.1384	0.1559	0.1591	0.1575	0.1496
	cw12b	0.1218	0.1359	0.1289	0.1153	0.1191	0.1172	0.0864	0.0843	0.0854	0.1742	0.1490	0.1616	0.1232
	cw09a	0.1620	0.1693	0.1656	0.1240	0.0959	0.1099	0.1626	0.1519	0.1572	0.1659	0.1421	0.1540	0.1467
	cw12a	0.1211	0.1338	0.1275	0.1286	0.1248	0.1267	0.1148	0.1078	0.1113	0.1648	0.1388	0.1518	0.1293
Grand Average		0.1457	0.1543	0.1500	0.1163	0.1103	0.1133	0.1241	0.1214	0.1227	0.1623	0.1472	0.1547	0.1352

than cw09b. Both of these observations can also be made from Figure 1: 1st row, 2nd column.

On the other hand, for ClueWeb12, some different observations arise than from ClueWeb09. In particular, for CW12A, there are no significant differences between the effectiveness of the trained models obtained from cw12a and cw12b. However, for CW12B, training on cw12a results in significantly more effective models - see Table 4.

To explain this surprising observation, which contrasts with that observed for ClueWeb09, recall that ClueWeb12 category B contains randomly sampled documents from the corresponding category A corpus. This being the case, many relevant documents (particularly those with higher relevance grades such as homepages) may be omitted from the smaller corpus. On the other hand, for ClueWeb09, category B contains the higher crawl priority documents within the larger category A. As crawling is inherently a costly process that cannot completely crawl the (infinite) Web, crawl priorities are used to target documents more likely to be relevant [27]. This is true for category B, which contains high crawl priority documents as well as Wikipedia. In contrast, ClueWeb09 category A reportedly contains a higher proportion of spam documents than category B [7].

To explain further the difference between the A and B categories of ClueWeb09 and ClueWeb12, Table 5 shows the number of relevant documents for the different relevance grades (e.g. relevant:  $\geq 1$ ; highly relevant:  $\geq 2$ ) in both categories of each corpus. From this, for ClueWeb09, it can be seen while category B is only 10% of the size of category A, it contains 58% of the relevant documents (relevance label 1 and above) and 65% of highly relevant documents (label 2 and above), with smaller percentages for document with higher relevance labels, such as perfect (label 4, 67%). In

contrast, for ClueWeb12, the category B (which is a 7% random sample of category A) only contains 20% of the relevant documents. While not high compared to ClueWeb09, 20% is higher than the 7% expectation, and can be explained by the fact that there were 9 TREC 2013 category B run submissions that contributed to the assessment pool [6].

Next, Table 5 also reports the statistics of the candidate document sets for each query obtained by BM25, from each of the category A and category B corpora, as well as the statistics of category B documents retrieved within the A candidate sets (denoted  $B \in A$ ). From this, we note that documents from the category B ClueWeb09 corpus form 12% of the retrieved documents in the A candidate sets (126,768 vs. 979,361) - this is in line with the relative proportion of ClueWeb09 category B within the larger category A corpus (10%). Going from the B candidate sets to the A candidate sets increases the number of relevant (labels  $\geq 1$ ) documents by 19% (31,995 to 36,101). However, the number of relevant documents retrieved in the A candidate sets that were also retrieved for B are reduced (31,995 to 20,499) - a trend that is mirrored across all relevance grades. Therefore, while more relevant documents are retrieved in ClueWeb09 category A candidate sets, they are less likely to be those from category B, and are also less likely to be perfectly relevant documents (relevance grade 4).

In contrast, for ClueWeb12, category B accounts for a far smaller proportion of the judged and relevant documents (25% and 20%, respectively) - and even sparser for higher relevance grades. This explains why training on cw12a generates better models for CW12B than cw12b does: there are more labelled documents in cw12a, and hence more effective and robust learned models can be obtained.

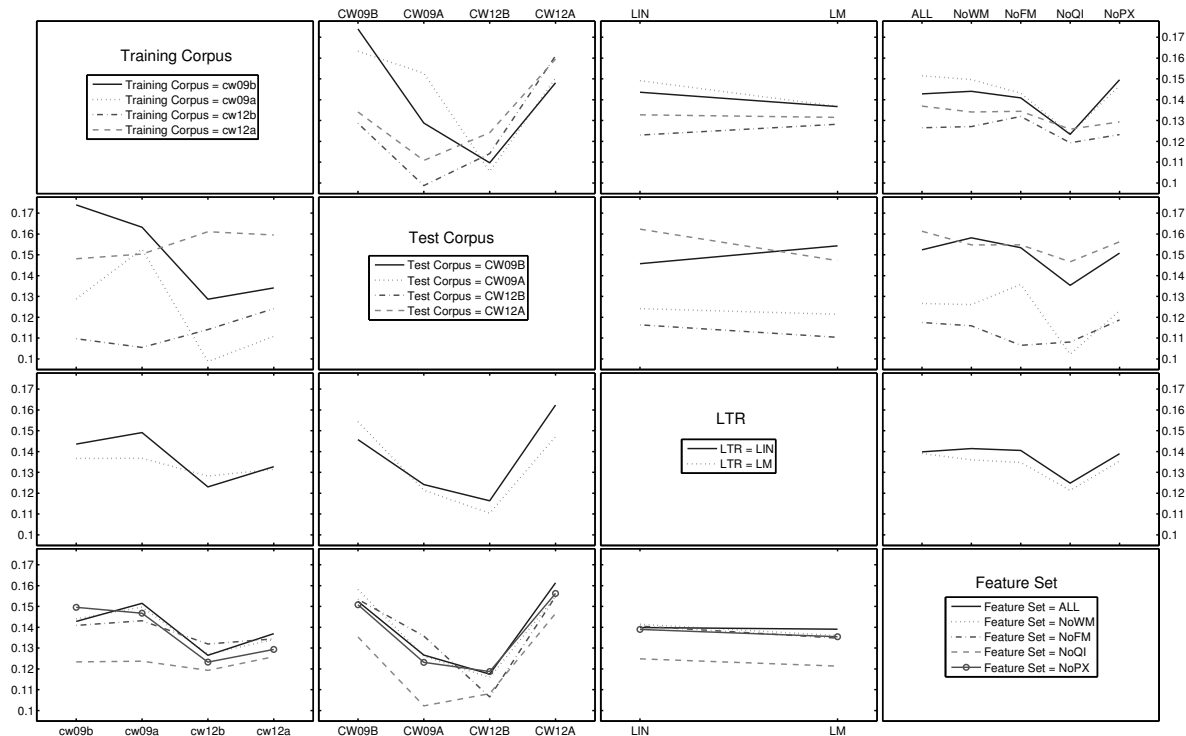


Figure 1: Interactions plot for the four factors (training corpus, test corpus, learning to rank technique and ablated feature set) in our full-factorial experiment.

From these statistics, we draw two conclusions: firstly, that the ClueWeb09 category B reflects an easier corpus, where perfect, highly relevant and relevant documents are easier to identify. This is reflected in the higher number of relevant documents retrieved in the candidate sets, as well as the higher ERR@20 scores obtainable on that corpus; Moreover, this makes cw09a unsuitable as a training corpus for learning effective models for CW09B, as the statistics of features for relevant documents identified on cw09a differ from cw09b; In contrast, for ClueWeb12, as the cw12a candidate document sets are more likely to contain high quality documents, it generates more effective learned models for use on CW12A than does cw12b. In summary, Null Hypothesis 1 cannot be rejected in general for within-corpus training. However, for CW12B there are insufficient labelled, relevant documents in the topics to obtain effective learned models, and hence within-corpus training from cw12a is appropriate.

## 4.2 Cross-Corpus Training

Similar to Section 4.1, this section also addresses Null Hypothesis 1, but for the cross-corpus transfer of learned models – i.e. from ClueWeb09 to ClueWeb12. For this section, we mostly focus upon the role of ClueWeb09 (cw09a and cw09b) in obtaining effective learned models for the newer ClueWeb12 corpus, c.f. CW12B and CW12A. For this analysis, we return to Table 4. From the table, it can be observed that for CW12A, the effectiveness of the models obtained from cw09a and cw09b are not significantly different from those obtained using cw12a or cw12b. This suggests that ClueWeb09 and ClueWeb12 are statistically indistinguishable as a training corpus for ClueWeb12. In contrast, for CW09A, models obtained from cw12a or cw12b are both

significantly less effective than those obtained from the target corpus itself. From Figure 1 the same observations can be made using the graph within the 1st column of the 2nd row. Within this graph, the lines associated with the ClueWeb12 test corpora are much flatter than those associated with ClueWeb09. This suggests that ClueWeb12 is less sensitive than ClueWeb09 to the change of training corpus, over all learning to rank techniques and feature sets.

Hence based on the significance tests observed in Table 4, Null Hypothesis 1 holds true for ClueWeb09 only, while for ClueWeb12, there is insufficient empirical evidence to reject it, meaning that there appears to be no significant effectiveness disadvantage in simply transferring learned models from ClueWeb09 to ClueWeb12 in general.

For the purpose of such a transfer, instead of using a learned model obtained from a different corpus, it is better to adapt the learned model being transferred to the new target corpus. In the context of the Yahoo! learning to rank challenge transfer task, Geurts and Louppe [12] examined six different methods of achieving transfer learning within learning to rank. Of these six methods, we note that the method that we call *model-feature transfer* was shown to be most effective. In this method, the output of the learned model on the older corpus taken as a new feature on the target corpus before re-learning. More formally, consider the predictions of a learned model  $\mathcal{M}$  obtained on a corpus  $c$  with features  $\mathcal{F}$  is denoted  $\mathcal{M}(c, \mathcal{F})$ . Then, to predictions using a feature transferred from  $c_1$  to  $c_2$  can be expressed as  $\mathcal{M}(c_2, \mathcal{F} + \mathcal{M}(c_1, \mathcal{F}))$ .

In Table 6, we report the effectiveness of model-feature transfer learning for CW12A. In particular, for CW12A we report the effectiveness of model-feature transfer, denoted

Table 5: Statistics of ClueWeb09 & ClueWeb12 category A and category B corpora and judgements, as well as corresponding statistics from the BM25 candidate document set. Some of these statistics were previously reported by [32] in the context of ClueWeb09 and the TREC 2009 Web track only. Recall that category B corpora are subset of the corresponding category A (10% for ClueWeb09, 7% for ClueWeb12).

Category	Crawled	Judged	$\geq 1$	$\geq 2$	$\geq 3$	$= 4$
relevance assessments						
CW09A	503,903,810	81,520	18,771	5,675	1,456	858
CW09B	50,220,423 10%	50,593 62%	11,037 58%	3,719 66%	895 61%	580 67%
of which retrieved by BM25						
CW09A	979,361	36,102	10,935	3,412	881	491
CW09B $\in$ CW09A	126,768	20,499	6,177	2,124	493	302
CW09B	972,049	31,995	8,959	3,096	774	498
relevance assessments						
CW12A	732,601,381	14,474	4,150	1,106	186	7
CW12B	52,315,578 7.1%	3,668 25%	829 20%	193 17%	20 11%	0 0%
of which retrieved by BM25						
CW12A	250,000	9,066	3,107	838	114	7
CW12B $\in$ CW12A	17,698	2,008	636	144	11	0
CW12B	250,000	2,647	786	188	20	0

Table 6: Comparison of model-feature transfer learning on ClueWeb12 - the model learned cw09a is used to create a supplemental feature used when training and ranking on cw12a. Significant differences according to the paired t-test compared to the model learned cw09a are denoted by  $\dagger$ .

Test Corpus	Feature Set	LIN		LM	
		cw12a	cw09a+ cw12a	cw12a	cw09a+ cw12a
CW12A	ALL	<b>0.1724</b>	0.1694	0.1680	<b>0.1682</b>
	NoPX	<b>0.1648</b>	0.1630	0.1388	<b>0.1721<math>\dagger</math></b>
	NoQI	<b>0.1580</b>	0.1574	0.1628	0.1479
	NoFM	0.1595	<b>0.1663</b>	0.1480	<b>0.1544</b>
	NoWM	<b>0.1603</b>	0.1537	0.1623	<b>0.1672</b>
	Average	<b>0.1630</b>	0.1620	0.1560	<b>0.1620</b>

cw09a + cw12a in contrast to learning on cw12a alone. On analysis of the results in Table 6, we note that model-feature technique benefits the effectiveness of the LambdaMART learning to rank technique on both CW12A. Indeed, while on average AFS is more effective than LambdaMART on CW12A without transfer learning (0.1630 vs. 0.1560), supplementing cw12 with a transfer learning feature from cw09a increases LambdaMART’s effectiveness to average 0.1620.

The improvements of LambdaMART is because regression trees need significant training data - particularly to recover linear relationships between a feature value and relevance [13, Chapter 9] such as between BM25 and relevance. By using transfer learning from a corpus with more labelled documents, the linear relationship between (say) BM25 and relevance can be better learned on the older corpus, and the model then fine-tuned on the newer corpus. Indeed, this is illustrated in Figure 2, which shows a partial dependence plot [13] of the LambdaMART predictions for learned models obtained from cw09a and cw12a as BM25 is varied. From the figure, it can be observed that the cw09a model has more decisions points for BM25 within its regression tree than that obtained from learning on cw12a (18 vs. 12).

On the other hand, for AFS, model-feature transfer learning does not benefit effectiveness, as the AFS model is unable to successfully encapsulate the transferred feature. Indeed, on inspection of the learned models obtained for the ALL

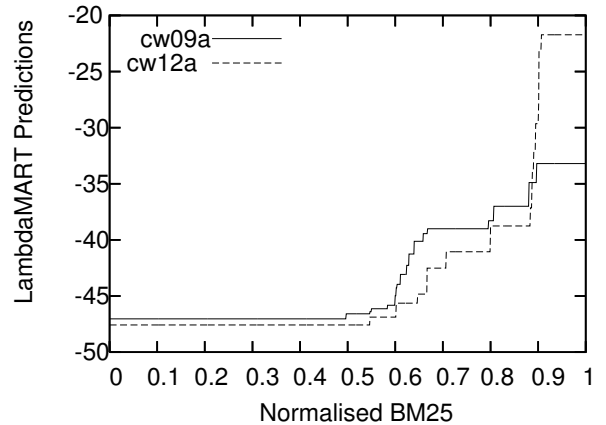


Figure 2: Partial dependence plots showing how predictions from LambdaMART change as a function of BM25, for models trained on cw09a and cw12a.

feature set, we find that the model trained on cw12a that encapsulates the transferred feature from cw09a actually used more features than the model obtained from cw12a alone. This suggests that AFS is actually trying – unsuccessfully – to ‘undo’ the work of the AFS model from cw09a to better adapt it to cw12a.

In summary, we find that Null Hypothesis 1 for cross-corpus training holds when targeting ClueWeb09 (see Table 4). However, for ClueWeb12, the available empirical evidence is not as strong as for ClueWeb09, as models trained on ClueWeb12 are not significantly more effective for this corpus than models obtained from ClueWeb09. This means that a learned model from ClueWeb09 can be directly applied for retrieval on ClueWeb12 with statistically comparable effectiveness. On the other hand, cross-corpus training can significantly benefit effectiveness on ClueWeb12 when the training sets from ClueWeb09 and ClueWeb12 are combined using the model-feature transfer learning method defined in [12] (see Table 6).



**Table 7: Average ERR@20 scores, factored out w.r.t. test corpora (rows) vs. Feature Set (columns), over all training corpora and learning to rank techniques. † on a score indicates that paired *t*-test gives significance to the difference between that score and the score the associated method achieved with all features ( $p < 0.025$ ).**

Test Corpus	Feature Set					Average
	ALL	NoWM	NoFM	NoQI	NoPX	
CW09B	0.1523	<b>0.1581</b>	0.1534	0.1353†	0.1508	0.1500
CW09A	0.1267	0.1260	<b>0.1357†</b>	0.1022†	0.1231	0.1227
CW12B	0.1174	0.1159	0.1065†	0.1081	<b>0.1187</b>	0.1133
CW12A	<b>0.1614</b>	0.1547	0.1547	0.1467	0.1562	0.1547
Average	<b>0.1395</b>	0.1387	0.1376	0.1231	0.1372	0.1352

### 4.3 Feature Sets

Next, we move onto our second null hypothesis, concerning the importance of feature sets. Table 7 reports the observed average ERR@20 scores for the ClueWeb09 and ClueWeb12 test corpora with varying feature sets over all training and learning to rank techniques. This table corresponds to the graph in the 2nd row, 4th column of Figure 1. Recall that as we are performing an ablation study, we analyse the importance of a set of features by observing its impact when removed from the learned model - e.g. NoFM denotes when the FM feature set (field-based weighting models) is removed (ablated) from the ALL feature set.

Multiple comparisons of the possible combinations of the levels of the training corpus, feature set, factors and learning to rank technique under consideration can be made based on the Friedman’s test, as shown in Figures 3 & 4 for CW09A and CW12A, respectively. Friedman’s test is the nonparametric counterpart of the balanced two-way ANOVA test: it tests for row effect (i.e., runs) after adjusting for possible column effects (i.e., queries). Hence, it is more appropriate for the multiple comparisons of the results of IR experiments than ANOVA and its two-sample Student’s *t*-test counterpart [14]. Within Figures 3 & 4, each group represents a different combination of learning to rank technique, and training corpus, where the feature sets are varied within the groups. Hence, the figures show the pairwise comparisons that are made across all training corpora and feature sets within Table 7.

On analysing Table 7, we note that removing the query independent features (i.e. NoQI) consistently harms the effectiveness obtained on both ClueWeb09 and ClueWeb12. For CW09B and CW09A, this impact is statistically significant relative to the effectiveness obtained using the ALL feature set. Indeed, referring to Figure 3, we observe significant decreases in effectiveness in removing the QI features (as the confidence intervals do not overlap) except when LambdaMART is trained on cw09b. For CW12B and CW12A - shown in Figure 4, although the loss in effectiveness with respect to the ALL feature set is not statistically significant (possibly due to the fewer topics available for ClueWeb12), removing QI causes more than a 10% decrease in absolute ERR@20 effectiveness. We believe the major difference between ClueWeb09 and ClueWeb12 in this respect is as follows: while ClueWeb09 category A corpus contains many spam documents [7], spam removal was conducted on ClueWeb12<sup>5</sup>, which will likely reduce the importance of the query independent features, many of which are intended

to identify low quality documents (e.g. spam classification score). Overall, our QI results suggest that, for the AFS and LambdaMART learning to rank techniques, and for all of the training corpora under consideration, query independent features provide, on average, a major contribution to the effectiveness of the learned models.

In general, as shown in the 4th row, 3rd column of Figure 1, the importance of feature sets is independent of the choice of learning to rank technique, since the lines that correspond to the various feature sets are horizontal over the learning to rank techniques. The difference in the effectiveness of NoQI and the other feature sets again shows us that removing QI features significantly reduces the effectiveness of both learning to rank techniques (see Table 7 & Figure 3).

The case of the field-based weighting models (FM) is different from that of QI, in that removing the query dependent FM features makes a significant gain in the effectiveness of the learned models for CW09A while, in contrast, it causes a significant loss in effectiveness for CW12B (this can also be observed in Figure 1: 2nd row, 4th column, and a marked loss for CW12A. In particular, while the query dependent features in the WM set encapsulate the anchor text, only the FM feature set allows the learner to separately weight the presence of query terms within the anchor text. We believe that these results suggest that the presence of spam within ClueWeb09 (particularly category A) can mislead the learner as to the usefulness of the anchor text - which will vary according to the prevalence of spam in different queries. On the other hand, with the reduced amount of spam in ClueWeb12, the FM feature set is useful for retrieval, and its ablation results in effectiveness degradations, which are significant in the case of CW12B.

In summary, as we have shown that QI and FM feature sets exhibit different effectiveness benefits between the CW09A and CW12B corpora, we conclude that Null Hypothesis 2 can be rejected for these feature sets, meaning that not all the feature sets contribute uniformly to the effectiveness of the learned models across different corpora. This emphasises the importance of appropriate training on the ClueWeb12 corpus, for instance using the model-feature transfer learning method that was investigated in Section 4.2.

## 5. CONCLUSIONS

This paper studies the generalisation and transferability of learning to rank models using the TREC ClueWeb09 and ClueWeb12 corpora as well as the contrasting the usefulness of different types of features - both within subsets of the same corpus, and across corpora. We formulated these research investigations as two null hypotheses, and conducted a thorough full-factorial experimental design using 250 TREC Web track topics, to derive empirically justified best practices for effective retrieval on the ClueWeb12 corpus. Indeed, our experimental results surprisingly suggest that the transfer of learned models from ClueWeb09 are sufficient for effective retrieval on ClueWeb12. However, the supplemental use of ClueWeb09 training data within the ClueWeb12 learning process can further significantly improve the effectiveness of learned models from the regression-tree based LambdaMART learning to rank technique.

We also found that the category B random subset of the ClueWeb12 corpus has insufficient labelled documents to obtain effective learned models. This contrasts greatly with the category B controlled subset of ClueWeb09, which contains 25% of the relevant documents within category A, despite

<sup>5</sup>See <http://lemurproject.org/clueweb12/specs.php>



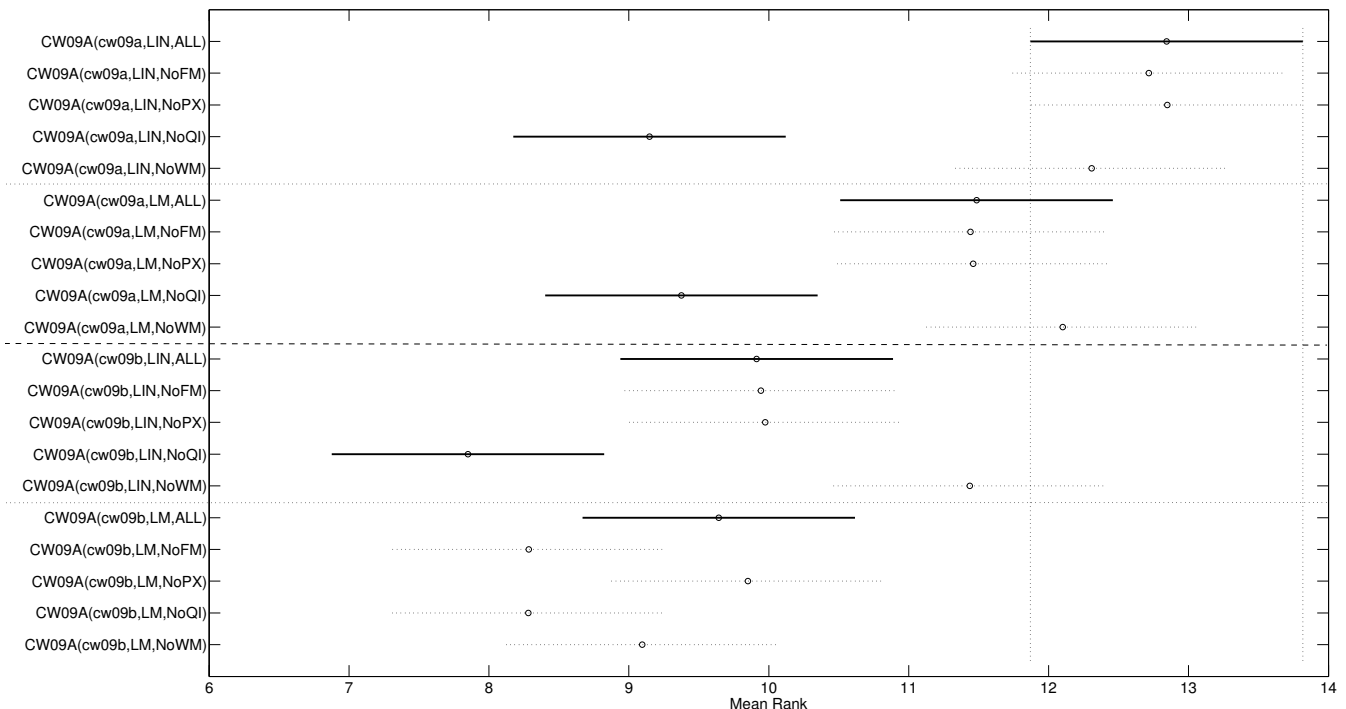


Figure 3: Multiple comparisons for combinations of levels for 3 factors under consideration (training corpus, feature set and learner) for CW09A. The circles show the mean ranks associated with each run in the column and the horizontal lines crossing the circles represent the 95% confidence interval (CI) for the mean ranks. Non-overlapping CIs indicate significant differences.

being only 10% in size. We conclude that the random sampling of ClueWeb12 category B from the larger corpus reduces the experimental value of this corpus, as many queries do not have highly relevant documents within this subset.

Lastly, we empirically showed the value of query independent features, as our results show that – irrespective of the learning to rank technique and the training corpus – every setting of ClueWeb09 requires their presence for effective retrieval. In contrast, the value of query independent features is less strong for ClueWeb12. Moreover, while adding various field-based weighting models as features could add effectiveness for ClueWeb12, their value was less apparent on ClueWeb09 which comparatively has more spam documents than ClueWeb12.

Similar to [4], we believe that the generalisation and transferability of learning to rank models are of significant importance, and hence this paper illustrates how researchers and practitioners must consider any biases present within corpora when conducting transfer learning, and how this may impact upon the usefulness of different types of features. In the future, we aim to adapt existing instance-based transfer learning techniques to the learning to rank scenario, and also investigate how risk-sensitive retrieval can be utilised within a transfer learning setting.

## 6. REFERENCES

- [1] G. Amati, E. Ambrosi, M. Bianchi, C. Gaibisso, and G. Gambosi. FUB, IASI-CNR and University of Tor Vergata at the TREC 2007 Blog track. In *Proceedings of TREC*, 2007.
- [2] M. Bendersky, W. B. Croft, and Y. Diao. Quality-biased ranking of Web documents. In *Proceedings of WSDM*, 2011.
- [3] C. J. Burges. From RankNet to LambdaRank to LambdaMART: An Overview. Technical Report MSR-TR-2010-82, Microsoft Research, 2010.
- [4] O. Chappelle and Y. Chang. Yahoo! learning to rank challenge overview. In *Proceedings of Yahoo! Learning to Rank Challenge*, 2011.
- [5] C. L. A. Clarke, N. Craswell, and I. Soboroff. Overview of the TREC 2012 Web track. In *Proceedings of TREC*, 2012.
- [6] K. Collins-Thompson, P. Bennett, F. Diaz, C. L. A. Clarke, and E. Voorhees. TREC 2013 Web track overview. In *Proceedings of TREC*, 2013.
- [7] G. V. Cormack, M. D. Smucker, and C. L. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Information Retrieval Journal*, 14(5), 2011.
- [8] N. Craswell, D. Fetterly, M. Najork, S. Robertson, and E. Yilmaz. Microsoft research at TREC 2009. In *Proceedings of TREC*, 2009.
- [9] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu. Boosting for transfer learning. In *Proceedings of ICML*, 2007.
- [10] V. Dang, M. Bendersky, and W. B. Croft. Two-stage learning to rank for information retrieval. In *Proceedings of ECIR*, 2013.
- [11] B. T. Dinçer, I. Kocabas, and B. Karaoglan. IRRa at TREC 2010: Index term weighting by divergence from independence model. In *Proceedings of TREC*, 2010.
- [12] P. Geurts and G. Louppe. Learning to rank with extremely randomized trees. In *Proceedings of Yahoo! Learning to Rank Challenge*, 2011.
- [13] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.
- [14] D. Hull. Using statistical testing in the evaluation of retrieval experiments. In *Proceedings of SIGIR*, 1993.
- [15] T. Kamishima, M. Hamasaki, and S. Akaho. Trbag: A simple transfer learning method and its application to

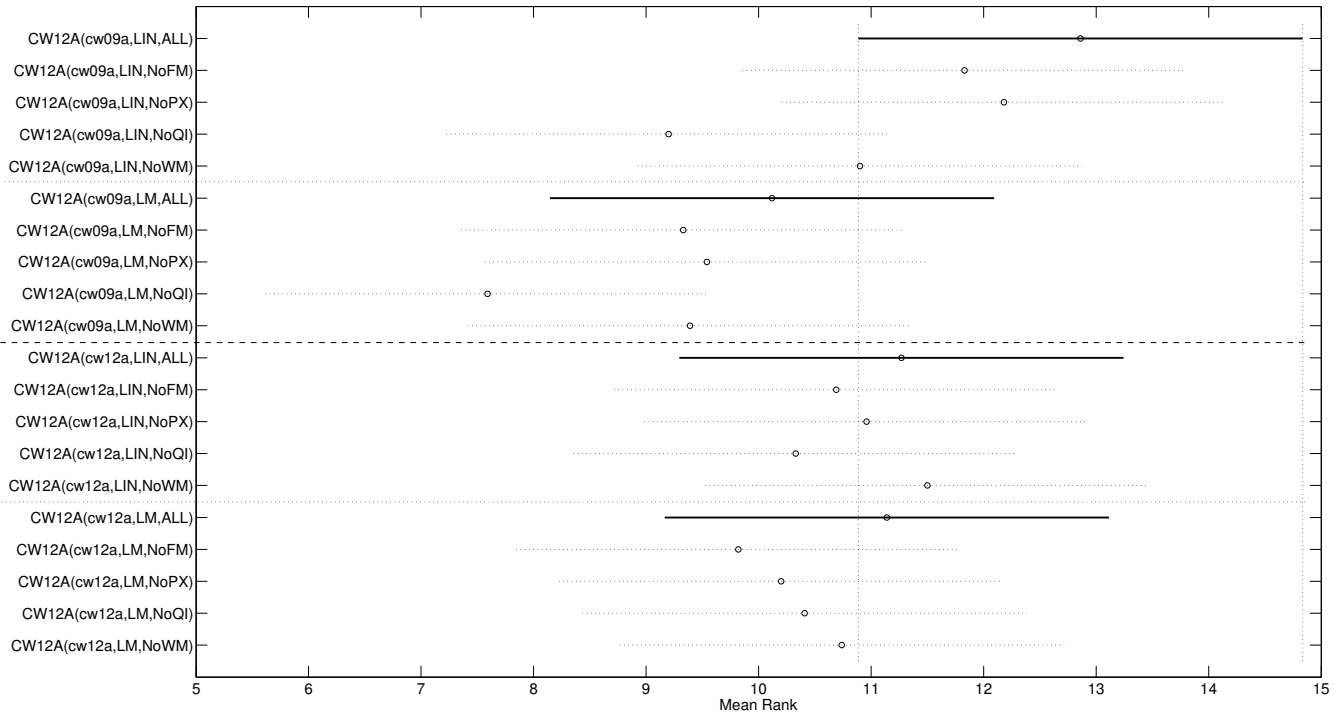


Figure 4: Multiple comparisons for combinations of levels for 3 factors under consideration (training corpus, feature set and learner) for CW12A. The circles show the mean ranks associated with each run in the column and the horizontal lines crossing the circles represent the 95% confidence interval (CI) for the mean ranks. Non-overlapping CIs indicate significant differences.

- personalization in collaborative tagging. In *Proceedings of ICDM*, 2009.
- [16] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science Journal*, 220(4598), 1983.
- [17] I. Kocabas, B. T. Dinger, and B. Karaoglan. A nonparametric term weighting method for information retrieval based on measuring the divergence from independence. *Information Retrieval Journal*, 17(2), 2014.
- [18] T.-Y. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval Journal*, 3(3), 2009.
- [19] T.-Y. Liu, T. Qin, J. Xu, W. Xiong, and H. Li. LETOR: Benchmark dataset for research on learning to rank for information retrieval. In *Proceedings of LR4IR at SIGIR*, 2007.
- [20] C. Macdonald, R. McCreadie, R. Santos, and I. Ounis. From puppy to maturity: Experiences in developing Terrier. In *Proceedings of OSIR at SIGIR*, 2012.
- [21] C. Macdonald, V. Plachouras, B. He, C. Lioma and I. Ounis. University of Glasgow at WebCLEF 2005: Experiments in per-field normalisation and language specific stemming. In *Proceedings of CLEF*, 2005.
- [22] C. Macdonald, R. Santos, and I. Ounis. The whens and hows of learning to rank for web search. *Information Retrieval Journal*, 16(5), 2012.
- [23] C. Macdonald, R. L. Santos, I. Ounis, and B. He. About learning models with multiple query-dependent features. *Transactions on Information Systems Journal*, 31(3), 2013.
- [24] D. Metzler. Automatic feature selection in the markov random field model for information retrieval. In *Proceedings of CIKM*, 2007.
- [25] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *Proceedings of SIGIR*, 2005.
- [26] T. Minka and S. Robertson. Selection bias in the LETOR datasets. In *Proceedings of LR4IR at SIGIR*, 2008.
- [27] M. Najork and J. L. Wiener. Breadth-first crawling yields high-quality pages. In *Proceedings of WWW*, 2001.
- [28] S. J. Pan and Q. Yang. A survey on transfer learning. *Transactions on Knowledge and Data Engineering Journal*, 22(10), 2010.
- [29] S. J. Pan, Q. Yang, and W. Fan. Tutorial: Transfer learning with applications. In *Proceedings of IJCAI*, 2013.
- [30] J. Peng, C. Macdonald, B. He, V. Plachouras, and I. Ounis. Incorporating term dependency in the DFR framework. In *Proceedings of SIGIR*, 2007.
- [31] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, A. Gull, and M. Lau. Okapi at TREC. In *Proceedings of TREC*, 1992.
- [32] R. L. Santos, C. Macdonald, and I. Ounis. Effectiveness beyond the first crawl tier. In *Proceedings of CIKM*, 2011.
- [33] S. Tyree, K. Q. Weinberger, K. Agrawal, and J. Paykin. Parallel boosted regression trees for web search ranking. In *Proceedings of WWW*, 2011.